ARMY RESEARCH LABORATORY

An Analysis of Clustering Tools for Moving Target Indication

by Anthony Martone, Roberto Innocenti, and Kenneth Ranney

ARL-TR-5037   **November 2009**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

An Analysis of Clustering Tools for Moving Target Indication

Anthony Martone, Roberto Innocenti, and Kenneth Ranney
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE:	3. DATES COVERED (From - To)
November 2009	Final	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Analysis of Clustering Tools for Moving Target Indication	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Anthony Martone, Roberto Innocenti, and Kenneth Ranney	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: RDRL-SER-U 2800 Powder Mill Road Adelphi, MD 20783-1197	ARL-TR-5037

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Previously, we developed a moving target indication (MTI) processing approach to detect and track slow-moving targets inside buildings, which successfully detected moving targets (MTs) from data collected by a low-frequency, ultrawideband radar. Our MTI processing algorithms include change detection (CD), used to identify the MT signature; automatic target detection (ATD), used to eliminate imaging artifacts and potential false alarms due to target multi-bounce effects; clustering, used to identify a centroid for each cluster in the ATD output images; and tracking, used to establish a trajectory of the MT. These algorithms can be implemented in a real-time or near-real-time system; however, a person-in-the-loop is needed to select input parameters for the clustering algorithm. Specifically, the number of clusters to input into the cluster algorithm is unknown and requires manual selection. A critical need exists to automate all aspects of the MTI processing formulation. In this report, we investigate two techniques that automatically determine the number of clusters: the knee-point (KP) algorithm and the recursive pixel finding (RPF) algorithm. The KP algorithm is a well-known heuristic approach for determining the number of clusters. The RPF algorithm is analogous to the image processing, pixel labeling procedure. Both routines processed data collected by our low-frequency, ultrawideband radar and their results are compared.

15. SUBJECT TERMS

moving target indication, impulse radar, UWB radar, change detection, clustering, k-Means clustering, knee-point cluster analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Anthony Martone
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	34	19b. TELEPHONE NUMBER *(Include area code)* (301) 394-2531
Unclassified	Unclassified	Unclassified			

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Current and future forces operating in urban environments need the capability to detect slow-moving personnel inside buildings. To identify moving personnel inside buildings, we consider a time-domain approach that uses a low-frequency, ultrawideband (UWB) radar. A low-frequency, UWB radar is desired since the low-frequency transmit pulse is capable of penetrating through the wall (*1*) and the UWB corresponds to a high range resolution that gives the capability to better locate the moving target (MT). In previous research (*2, 3*), we demonstrated the effectiveness of our time-domain, moving target indication (MTI) approach for detecting moving personnel inside wood and cinderblock structures, moving personnel walking in nonlinear trajectories, and multiple moving personnel walking in linear trajectories.

We consider a time-domain approach to MTI as an alternative to a frequency-domain approach, i.e., Doppler processing, since a very small Doppler shift in backscattered frequency is generated due to (1) the slow motion of the mover and (2) the low frequency needed to penetrate through the wall. Our time-domain processing algorithms are based on the change detection (CD) paradigm, which is inherently similar to clutter cancellation (*4*). In the CD paradigm, the Synchronous Impulse Reconstructive (SIRE) radar remains stationary and generates a set of images for a region of interest (ROI). Each image in the set is formed every two-thirds of a second. The stationary objects in the building remain in the same location in each image; however, moving personnel will be at different locations. We can therefore detect the moving personnel by subtracting adjacent images in the set, thereby eliminating the stationary objects and identifying the MT signature.

Additional processing is needed to enhance the MT signature and includes a constant false alarm rate (CFAR) algorithm, morphological processing, k-Means clustering, and a tracking algorithm. CFAR and morphological processing are approaches used to eliminate imaging artifacts and potential false alarms due to target multi-bounce effects. The k-Means clustering algorithm is used to identify centroids for given input clusters, where the clusters are produced by the CFAR and morphological processing algorithms. Finally the tracker is used to establish a trajectory of the MT based on the input centroids.

The algorithms in our MTI processing formulation can be implemented in a real-time or near-real-time system; however, a person-in-the-loop is needed to select input parameters for the k-Means clustering algorithm. Specifically, the number of clusters input into the k-Means routine is unknown and requires manual selection. A critical need exists to automate all aspects of the MTI processing formulation, which requires using an additional processing routine to identify the number of clusters input into the k-Means algorithm. In this report, we investigate two techniques that automatically determine the number of clusters: the knee-point (KP) algorithm and the recursive pixel finding (RPF) algorithm. The KP algorithm is a well-known heuristic

approach for determining the number of clusters. The RPF algorithm is analogous to the image processing, pixel labeling procedure. We used both routines to process data collected by our low-frequency, UWB radar and compared their results.

2.   Synchronous Impulse Reconstruction Radar

The U.S. Army Research Laboratory (ARL) has developed a ground-based SIRE radar (*5*) to aid in the detection of concealed targets (*6*). The SIRE radar is an impulse-based, UWB imaging radar with a bandwidth covering 300 MHz to 3 GHz, a frequency range appropriate for sensing through the wall (STTW) applications (*1*). As is illustrated in figure 1, the SIRE radar employs 2 transmit antennas and 16 receiver antennas mounted in a wooden frame and attached to the top of a Ford Expedition. Two impulse transmitters are located at either end and slightly above the receive array. The transmitters fire in an alternating sequence—the left transmitter followed by the right. Each transmitter launches a sequence of low-power pulses, and reflected energy is integrated within each receive channel to achieve an acceptable signal-to-noise ratio (SNR). The SIRE radar constructs a high-resolution (0.056 m) downrange profile through novel, ARL-developed signal-processing techniques (*7*). We effectively buffer two downrange profiles from each receive channel, and the time required to assemble these profiles represents one frame of data. After buffering the data from one frame, we then collect another pair of downrange profiles from each receive channel for the next frame of data. The downrange swath measured by the radar extends from approximately 10 to 35 m, and the amount of processing time required for downrange profile reconstruction results in a low effective pulse repetition interval (approximately two-thirds of a second). We have leveraged the SIRE radar as part of an overall investigation of MTI phenomenology.
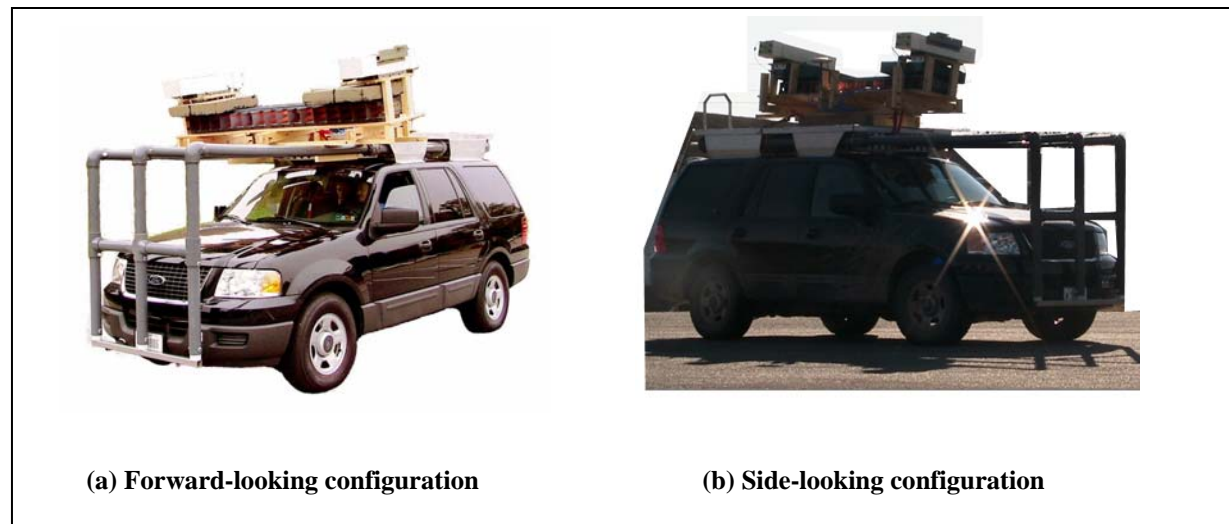

(a) Forward-looking configuration   **(b) Side-looking configuration**

Figure 1. The SIRE radar.

3. Moving Target Indication Methodology

In this section, we will discuss our MTI processing formulation. Our approach to MTI is shown in figure 2. The SIRE radar remains stationary and measures the energy reflected from an area under surveillance, or ROI. The 16 receive channels are processed using a time-domain, back-projection technique developed for synthetic-aperture-radar (SAR) image formation (*8*). The SAR image, output of the first block in figure 2, contains several artifacts making it difficult to identify the MT. A SAR image is generated for each frame of data collected by the SIRE radar, thereby forming a set of SAR images. Note that the SIRE radar processes each frame of data within two-thirds of a second, indicating the capability to extend our MTI processing formulation to a real-time or near-real-time solution.
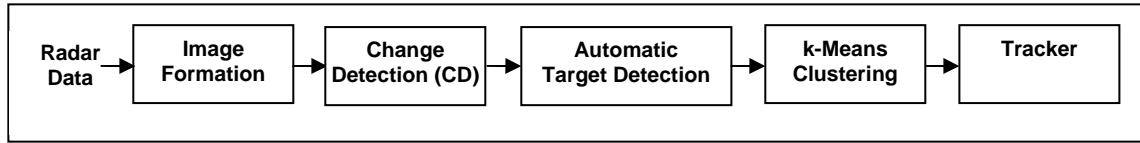
Figure 2. MTI processing formulation using data collected by the SIRE radar.

CD is applied to the SAR images, thereby creating a set of difference images. As discussed, the stationary objects in the building remain in the same position/location in each SAR image; however, moving personnel will be at different locations. We can therefore detect the moving personnel by applying CD, which subtracts subsequent SAR images, thereby eliminating the stationary objects and identifying the MT signature. To illustrate the effectiveness of the CD approach, consider the SAR images in figures 3(a) and (b). The SAR images are focused on the same target area at different moments in time, and one person is moving within the target area. As is evident, the SAR images contain several artifacts making it difficult to identify the moving target. The moving target is located by applying CD. The resulting difference image is shown in figure 3(c). It is clear from the difference image that most of the artifacts due to stationary clutter have been eliminated and the resulting MT signature is identified.

(a) SAR Image 1   **(b) SAR Image 2**   **(c) Difference Image**
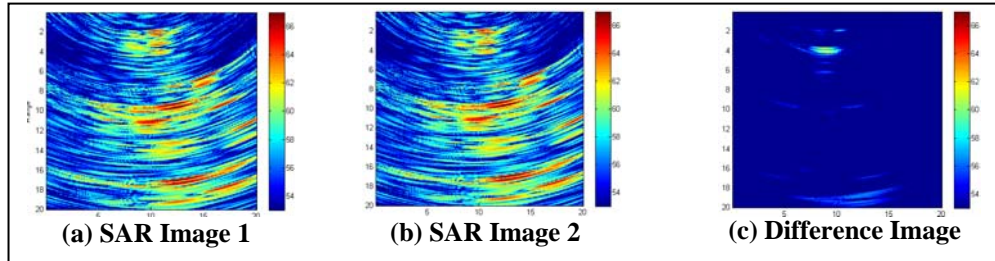
Figure 3. Example of two SAR images, (a) and (b), and the resulting difference image, (c). Notice that the MT signature is difficult to identify in the SAR images but is identified in the difference image.

Interpretation of the resulting MT signature is still challenging after CD. For example, CD cannot automatically identify the MT signature located in the difference image and the MT signature is only identified through visual inspection of a sequence of difference images. Therefore, it is not possible to implement additional signal processing techniques like classification using a single difference image. Another challenge with CD is that sidelobe artifacts are produced in the difference image, which confuse the true MT location.

A way to improve user interpretation of the resulting difference image is to apply automatic target detection (ATD) algorithms. The ATD algorithms considered are the CFAR algorithm and the morphological processing algorithm. CFAR is a unique approach to eliminating potential false alarms produced by slow-moving targets (*9*). The CFAR algorithm performs a test of local contrast that is designed to achieve a constant false alarm rate. It generates a CFAR image of binary pixels for each input difference image. This CFAR image contains clusters of points of interest (POIs) corresponding to either the MT signature or a false alarm. For example, consider the difference image and CFAR image of figure 4. The difference image is input into the CFAR algorithm and a CFAR image is output. The red cluster in the CFAR image corresponds to a group of POIs. This example illustrates that the CFAR algorithm identifies the MT signature and eliminates the sidelobe and other artifacts present in the difference image.

(a) Difference Image   **(b) CFAR Image**
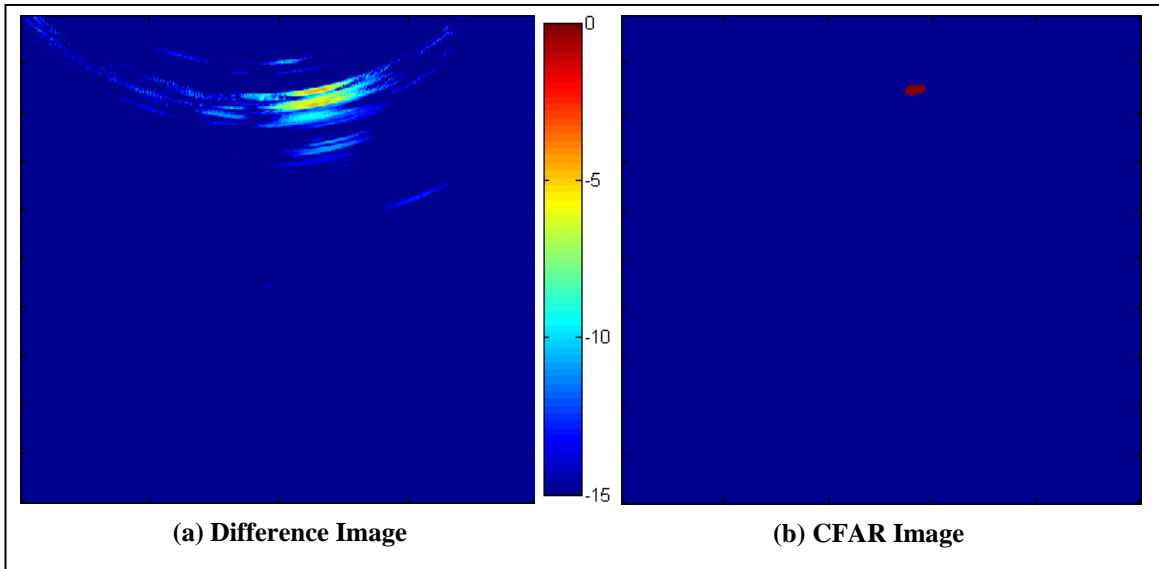
Figure 4.   The difference image is input into the CFAR algorithm and the CFAR image is output. The red cluster in the CFAR image corresponds to a group of POIs.

The POIs provide us with a list of possible MTs; however, some POIs correspond to false alarms. For example, consider the difference image and CFAR image shown in figure 5. As illustrated, the difference image contains true MT signatures and false MT signatures, i.e., false alarms. As is shown in the CFAR image, the CFAR algorithm does not eliminate all false alarms.
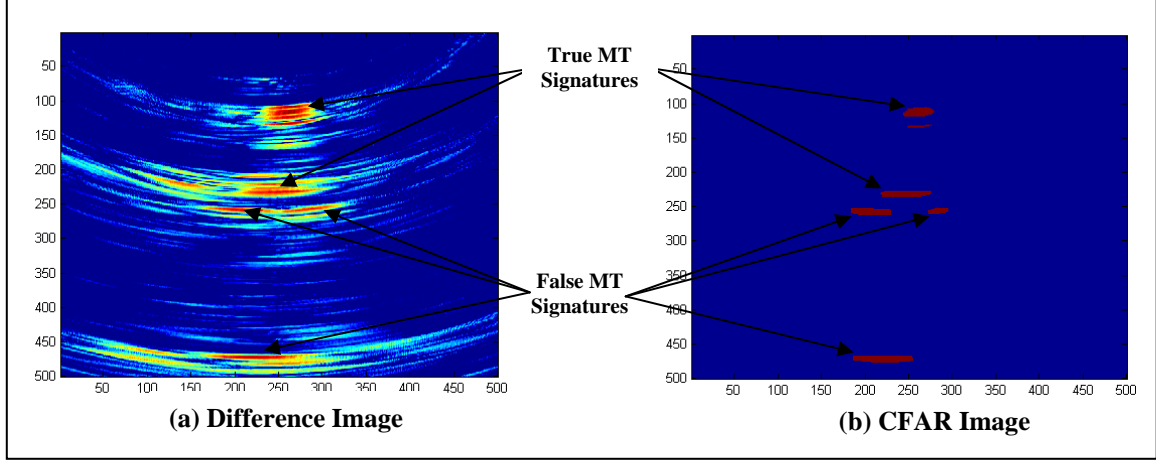
Figure 5. Example of a difference image and the resulting CFAR image. The CFAR image contains false MT signatures (i.e., false alarms).

One possible way to reduce the number of false alarms is to input the POIs into a morphological processing algorithm. The morphological processing implements a dilation and erosion procedure (*10*). Dilation is used to grow the POI clusters and erosion is used to shrink them. During the dilation process, POI clusters in close proximity, defined as 0.68 m, are connected. Erosion is then implemented to shrink the size of each POI cluster back to its original size. The morphological algorithm generates a morphological image for each input CFAR image.

Before the POIs are input into a tracking algorithm, their number must be reduced. This motivates the need to refine the number of POIs by using a clustering algorithm to identify centroids for each cluster of POIs. The clustering routine used by the MTI processing formulation of figure 2 is the well-known k-Means algorithm (*11, 12*). We will discuss the k-Means algorithm in detail in section 4.

The tracker algorithm (*13*) is another method used to reduce the number of false alarms. The centroids generated by the clustering algorithm serve as inputs to the tracker; so it is possible to have multiple tracker inputs even when a single moving target is present. These centroids may indicate the true position of a MT or false alarms. The tracker estimates the correlation between each centroid and the existing tracks and then associates the existing tracks with the most highly correlated (i.e., most reasonable) centroid. Non-assigned centroids are used to initiate new tracks and outdated tracks are deleted. A Kalman filter determines the present track position and predicts the next measurement.

4. Clustering Methodologies

In this section we describe the k-Means, KP, and RPF algorithms. The KP and RPF algorithms provide the k-Means algorithm with the number of clusters present in the morphological image. We define a cluster as a group of one or more connected POIs. We signify the total number of

clusters present in the morphological image as *T*. For example, consider the POIs in figure 6(a), where each POI corresponds to a blue diamond. The POIs connected to each other form a cluster as illustrated in figure 6(b). As is shown, *T=2* clusters are present in the morphological image. Once the number of clusters in the image is known, the k-Means algorithm is used to indicate the centroid of each cluster.

(a) Examples of POIs in the morphological image.   **(b) Non-disjoint POIs are clustered together.**
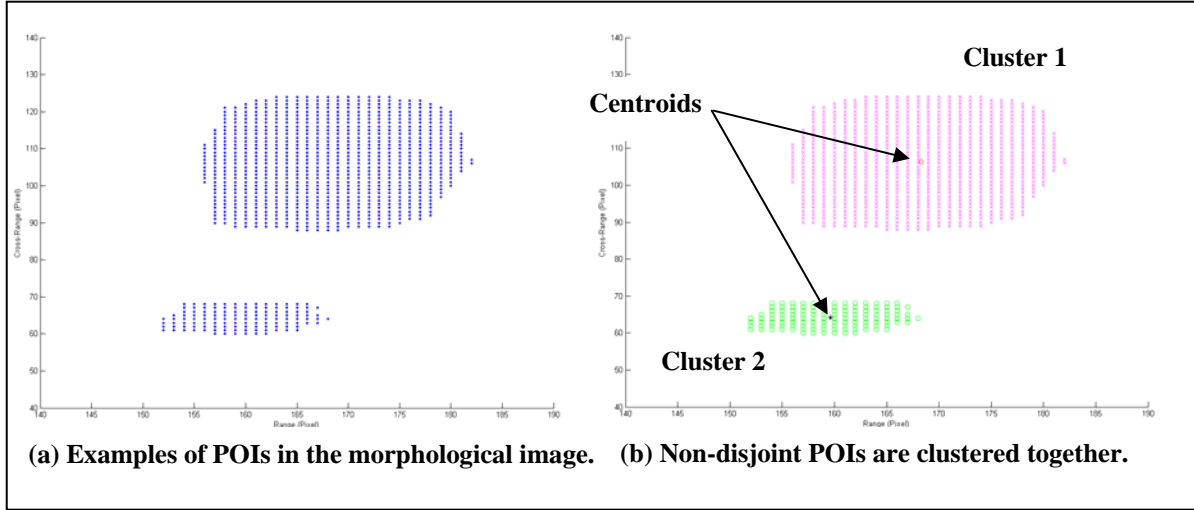
Figure 6. (a) The morphological image that contains POIs. When these images are input into a clustering algorithm, two clusters are identified; the clusters and corresponding centroids are shown in (b).

4.1   k-Means Algorithm

The clustering routine used by the MTI processing formulation of figure 2 is the well-known k-Means algorithm (*11, 12*). The k-Means algorithm identifies the centroids of the POIs by an iterative procedure. This iterative procedure minimizes the square-error between centroid estimates and their corresponding POIs. It should be noted that the clusters identified by the clustering algorithm are not unique and it is possible that the centroid locations differ for different iterations of the clustering algorithm. A block diagram of the k-Means algorithm is shown in figure 7.
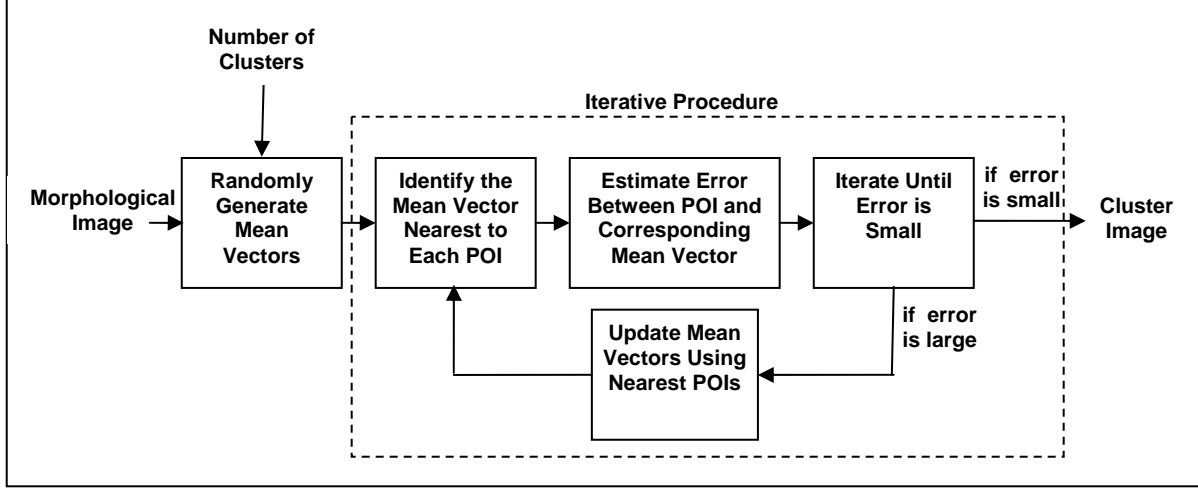
Figure 7. The k-Means algorithm.

The input morphological image contains a set of M POI vectors defined as

$$\{\rho_1, \ldots \rho_M\}, \tag{1}$$

where $\rho_i = [\hat{\rho}_{(i,X)} \quad \hat{\rho}_{(i,Y)}]$ is the ith POI vector, $\hat{\rho}_{(i,X)}$ is the cross-range component of ρ_i, and $\hat{\rho}_{(i,Y)}$ is the range component of ρ_i. The k-Means algorithm requires that the number of clusters, T, is known a priori. The k-Means algorithm begins by randomly generating T mean vectors defined as

$$\{\mu_1, \ldots \mu_T\}, \tag{2}$$

where $\mu_j = [\hat{\mu}_{(j,X)} \quad \hat{\mu}_{(j,Y)}]$ is the jth mean vector, $\hat{\mu}_{(j,X)}$ is the cross-range component of μ_j, and $\hat{\mu}_{(j,Y)}$ is the range component of μ_j. Note that T is unknown and must be manually defined. The KP and RPF algorithms automatically determine T. The mean vectors are considered as centroid estimates. The next step of the algorithm determines the mean vector nearest to each POI using the Euclidean distance measure:

$$D_{(i,j)} = \| \mu_j - \rho_i \|^2. \tag{3}$$

Next define the set S_j of size m_j to be the POIs closest to μ_j. Estimate the error between the mean and nearest POIs using the sum of squares (SOS) error criteria (14):

$$J_k = \sum_{j=1}^{T} \sum_{\rho_i \in S_j} \| \rho_i - \mu_j \|^2, \tag{4}$$

where J_k is the error for the kth iteration of the k-Means algorithm. This SOS error criterion is a measure of variance between the POI vectors and the nearest mean vectors and must be minimized. The following equation is used to determine if J_k is minimized:

7

$$| J_k - J_{k-1} | < \varepsilon , \qquad (5)$$

where J_{k-1} is the error for the (k–1) iteration, and ε is a threshold value. If the condition defined by equation 5 is satisfied, then the SOS error is minimized, thereby indicating the final centroid estimates represented by the mean vectors. If the condition defined by equation 5 is not satisfied, then an additional iteration is required and each mean vector is updated using its nearest POI:

$$\mu_j = \frac{1}{m_j} \sum_{\rho_i \in S_j} \rho_i . \qquad (6)$$

For example, consider the morphological image in figure 8. For this example, two mean vectors were randomly generated and indicated by the red diamond and black star. As the k-Means algorithm iterates, several centroids are estimated. Each newly generated estimate corresponds to a smaller SOS error.
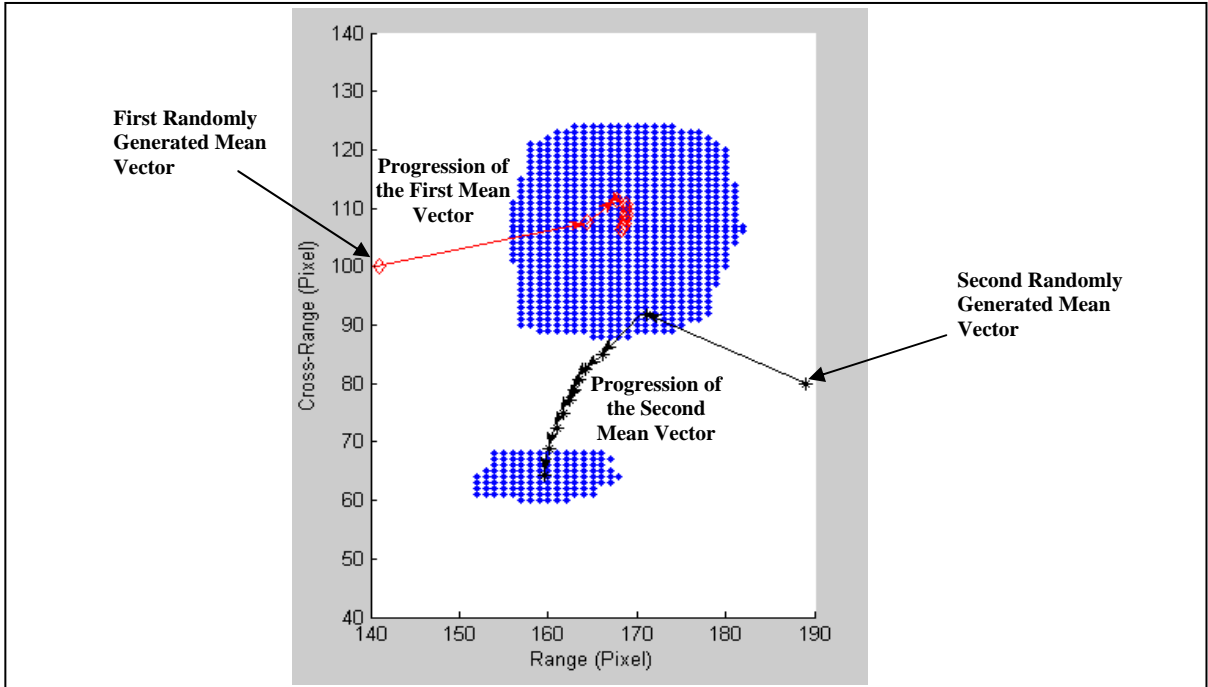
Figure 8. k-Means algorithm iteration example: the red line indicates progression of the first mean vector and the black line indicates progression of the second mean vector. Multiple iterations are needed to minimize the error between the POIs and nearest mean vectors.

4.2   KP Algorithm

The KP algorithm is one approach to automatically determine the number of clusters, T, present in the morphological image. The KP algorithm is a heuristic approach used to determine the optimal number of clusters (*15, 16*). The KP algorithm iterates the k-Means algorithm for many

different cluster number choices. This will produce a set of minimized SOS errors $\hat{J} = \left\{ \hat{J}_1, \hat{J}_2, \ldots \hat{J}_C \right\}$, where $\hat{J}_i$ is the ith minimized SOS error, and $i=1,2,\ldots C$ denotes the number of clusters. We normalize $\hat{J}$ by $\max(\hat{J})$ to obtain

$$\bar{J} = \left\{ \bar{J}_1, \bar{J}_2, \ldots \bar{J}_C \right\} = \hat{J} / \max(\hat{J}) \tag{7}$$

where $\bar{J}_i$ is the ith normalized minimized SOS (NMSOS) error. A heuristic used to identify the number of clusters searches for a large drop in NMSOS error, i.e., the "knee-point" (*15, 16*). For example, consider the images shown in figure 9*. By visual inspection it would appear that two clusters are present in the images (i.e., *T=2*). The NMSOS errors for each image of figure 9 are plotted in figure 10, where $N \in \left\{ 1, \ldots C \right\}$. As is shown in figure 10, a large gap exists between *N=1* and *N=2*, thereby indicating that the predicted number of clusters, N_p, is 2. $N_p=2$ correctly corresponds to the true number of clusters, *T*, in the images.
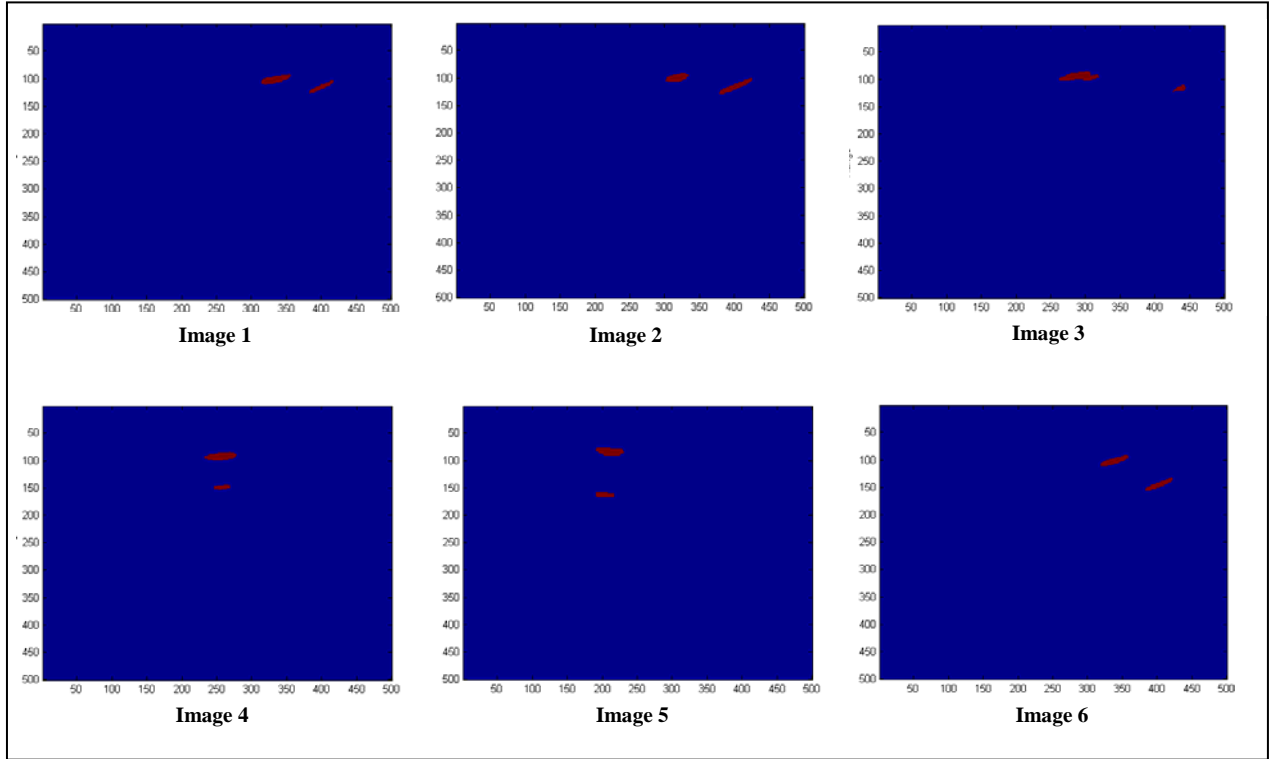

Figure 9. Images with POIs present. Through a visual inspection of the images, it would appear that two clusters are present (i.e., *T=2*).

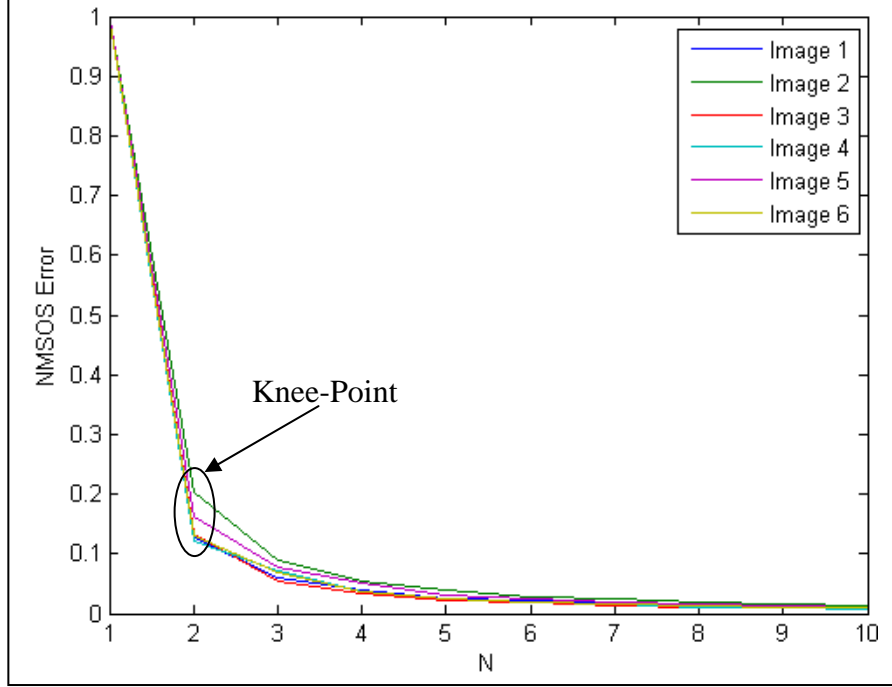* Images in example are CFAR images. This example/concept is easily extended for morphological images.

Figure 10. A plot of NMSOS errors for different values of N—a large gap exists between N=1 and N=2, thereby indicating the $N_p=2$ is the knee-point.

4.3 RPF Algorithm

Unlike the KP algorithm, the RPF algorithm does not implement the k-Means algorithm or rely on a heuristic to determine the number of clusters. Instead, the RPF algorithm scans the morphological image pixel-by-pixel for a POI. The POI is a binary (1 or 0) value and signifies that a cluster has been identified. The RPF algorithm then recursively scans every pixel of the cluster in a local area and identifies the cluster boundaries. This process is similar to a pixel labeling procedure. In pixel labeling (*10*), the image is scanned pixel-by-pixel from left-to-right, and top-to-bottom. A pixel is considered an "object" if the pixel equals 1, otherwise it is considered a "hole." All neighboring pixels of an object are examined and if a neighbor is an object, then it is merged with the object to which it is connected. The merged objects form a list that is used to identify all object pixels forming a single cluster.

A flow chart of the RPF algorithm is shown in figures 11 and 12. As is shown in figure 11, the morphological image *I(i,j)* is input into the RPF algorithm, where *0 < i < M* denotes range and *0 < j < Q* denotes cross-range. The RPF algorithm globally scans *I(i,j)* pixel by pixel from left-to-right and up-to-down. If *I(i,j)=0*, then the pixel coordinates are updated and the global scanning continues. If *I(i,j)=1*, then the pixel coordinates *(i,j)* are compared with a global list, *G_list*, of pixels. *G_list* is a list of pixel coordinates corresponding to previously identified pixel objects. If the current pixel at *(i,j)* is not in *G_list*, then it is considered a pixel object and *m=i*, *q=j*, and *(m,q)* are input into the function *ra()*.
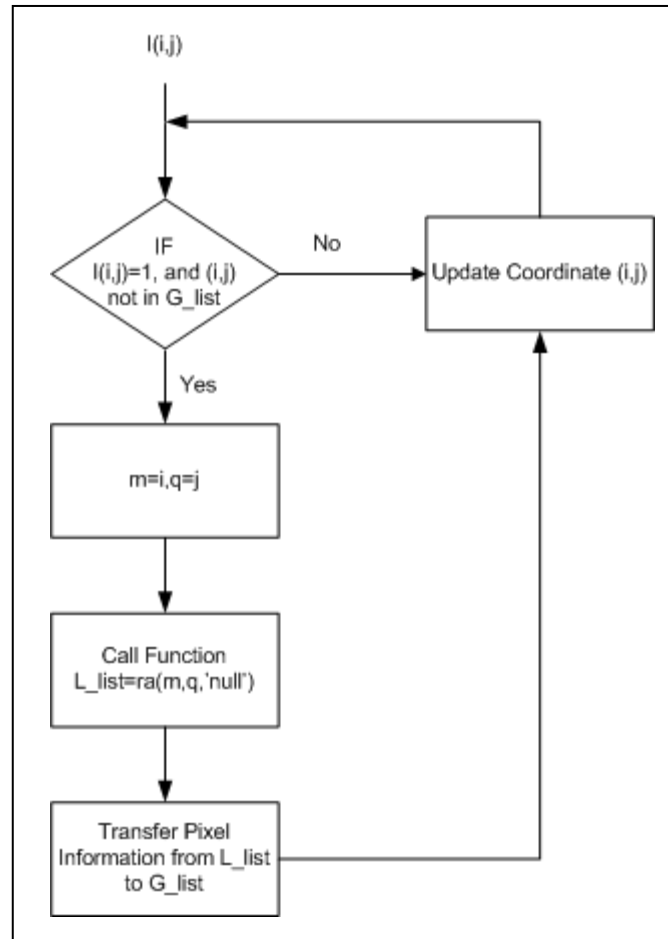
Figure 11. Global procedure of the RPF algorithm.
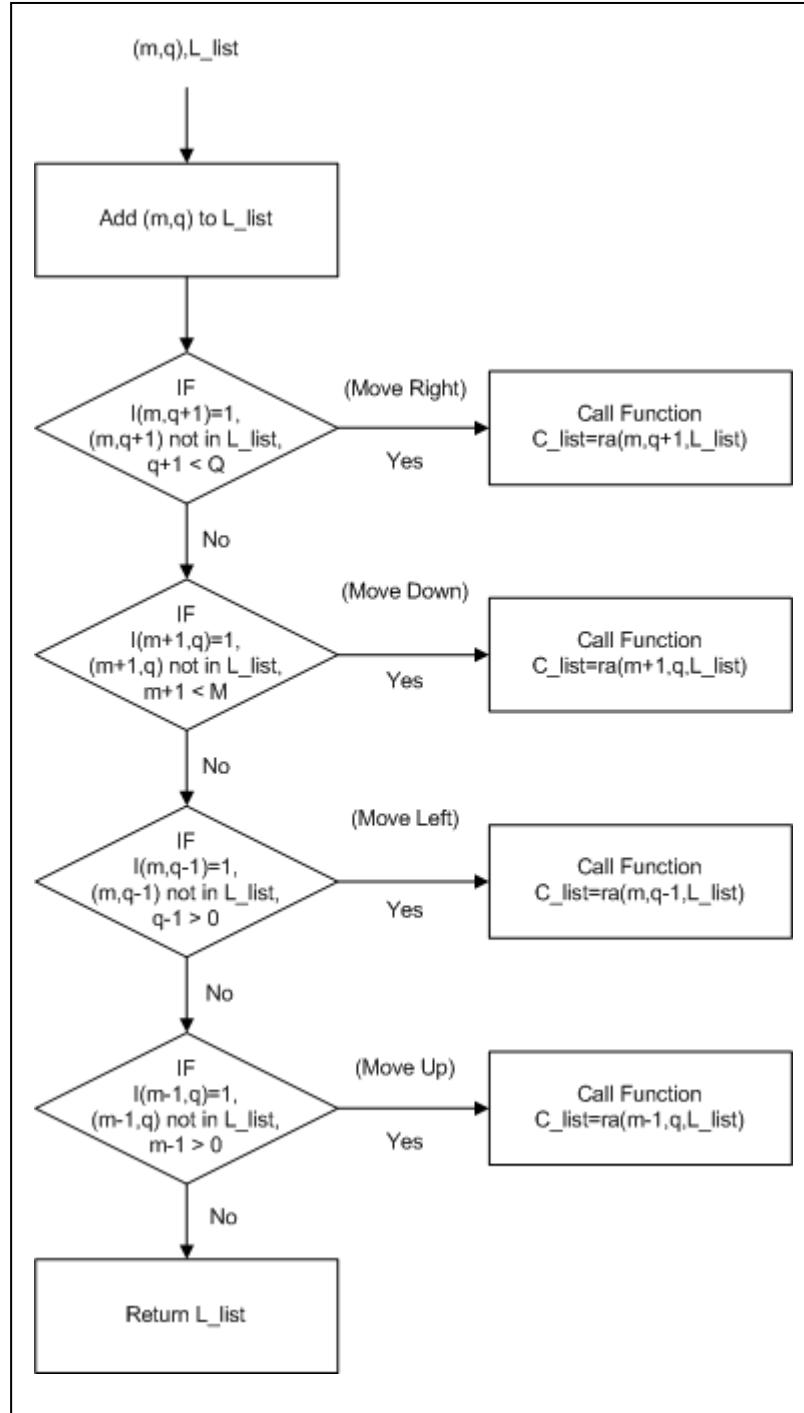
Figure 12. Local procedure for the RPF algorithm.

The function *ra()* is used to recursively scan the local area for additional pixel objects. The flow chart for the *ra()* algorithm is shown in figure 12. The inputs to the function *ra()* are the current pixel location *(m,q)* and a local list, *L_list*, of pixel objects (*L_list=<null>* for the first call). The function *ra()* first adds the pixel coordinates *(m,q)* to *L_list*. A neighboring pixel is then checked to determine if it is an object pixel. The neighboring pixel is also checked against boundary

conditions and *L_list*. If all conditions are satisfied, *ra()* is called again with a new pixel coordinate. This procedure continues until the last pixel in the cluster is scanned. *ra()* then returns *L_list* and the information from *L_list* is transferred to *G_list*. Pixel coordinate information is then updated and the global scan continues.

5. Experiments

Twelve data sequences of moving personnel were analyzed by the KP and RPF algorithms. This data consists of multiple scenarios (real time with no simulations) of personnel walking inside wood and cinderblock buildings. During data collection the SIRE radar remains stationary and is positioned broadside to the wall and 38° of the broadside angle, which was chosen due to practical considerations (expediting the experiment). The data sequences are described in table 1. "Images" correspond to the number of morphological images in the sequence. "Radar Angle" corresponds to the angle offset of the radar during collection. "Total Clusters" corresponds to the total number of clusters present in the sequence. "1C" to "6C" correspond to the number of morphological images in the sequence containing the specified number of clusters; for example, "6C" corresponds to the number of morphological images containing six clusters. "Description" corresponds to the mover's trajectory during the sequence. "Walk-Sit-Walk" describes a sequence where a person enters the room, sits in a chair, then stands and leaves the room. "Over-Take" corresponds to two people walking sequentially (i.e., one followed by the other) away from the radar where one person overtakes the other. "Random" corresponds to a person randomly walking inside the building. "Linear-Away" corresponds to a person walking in a linear direction away from the radar. "Opposite" corresponds to two people walking in opposite directions towards and away from the radar, where the two people cross paths.

Table 1. Description of the 12 data sequences of moving personnel analyzed by the KP and RPF algorithms.

	Building	Radar Angle	Description	Images	Total Clusters	1-C	2-C	3-C	4-C	5-C	6-C
1	Wood	0	Walk-Sit-Walk	98	161	53	28	16	1	0	0
2	Wood	0	Over-Take	35	76	13	8	10	3	1	0
3	Wood	0	Random	65	115	38	13	8	4	1	1
4	Wood	0	Linear-Away	45	90	16	19	7	1	1	1
5	Wood	0	Opposite	35	52	23	9	1	2	0	0
6	Wood	0	Walk-Sit-Walk	96	175	48	30	8	7	3	0
7	Wood	38	Circle	76	76	76	0	0	0	0	0
8	Wood	38	Walk-Sit-Walk	76	87	67	7	2	0	0	0
9	Cinderblock	0	Opposite	12	17	7	5	0	0	0	0
10	Cinderblock	0	Random	36	47	26	9	1	0	0	0
11	Cinderblock	38	Opposite	11	14	8	3	0	0	0	0
12	Cinderblock	38	Random	24	27	21	3	0	0	0	0
Total				609	937	396	134	53	18	6	2

5.1 KP Algorithm

The KP algorithm processed each frame of each data sequence described in table 1 to generate $\bar{J} = \{\bar{J}_1, \bar{J}_2, ... \bar{J}_C\}$, where $\bar{J}_i$ is again the ith NMSOS error and $C=6$. $C=6$ was chosen since the maximum number of clusters present in any morphological image is six. The NMSOS errors generated by the KP algorithm are illustrated in figure 13, where the NMSOS errors are graphically separated by T, the "true" number of clusters for a given morphological image. Additional statistics of the NMSOS errors were calculated and are shown in table 2, where "mean" corresponds to the mean of the NMSOS error and "std" corresponds to the standard deviation of the NMSOS error.

a) NMSOS errors for morphological images that contain T=1 clusters.

b) NMSOS errors for morphological images that contain T=2 clusters.

c) NMSOS errors for morphological images that contain T=3 clusters.

d) NMSOS errors for morphological images that contain T=4 clusters.

e) NMSOS errors for morphological images that contain T=5 clusters.

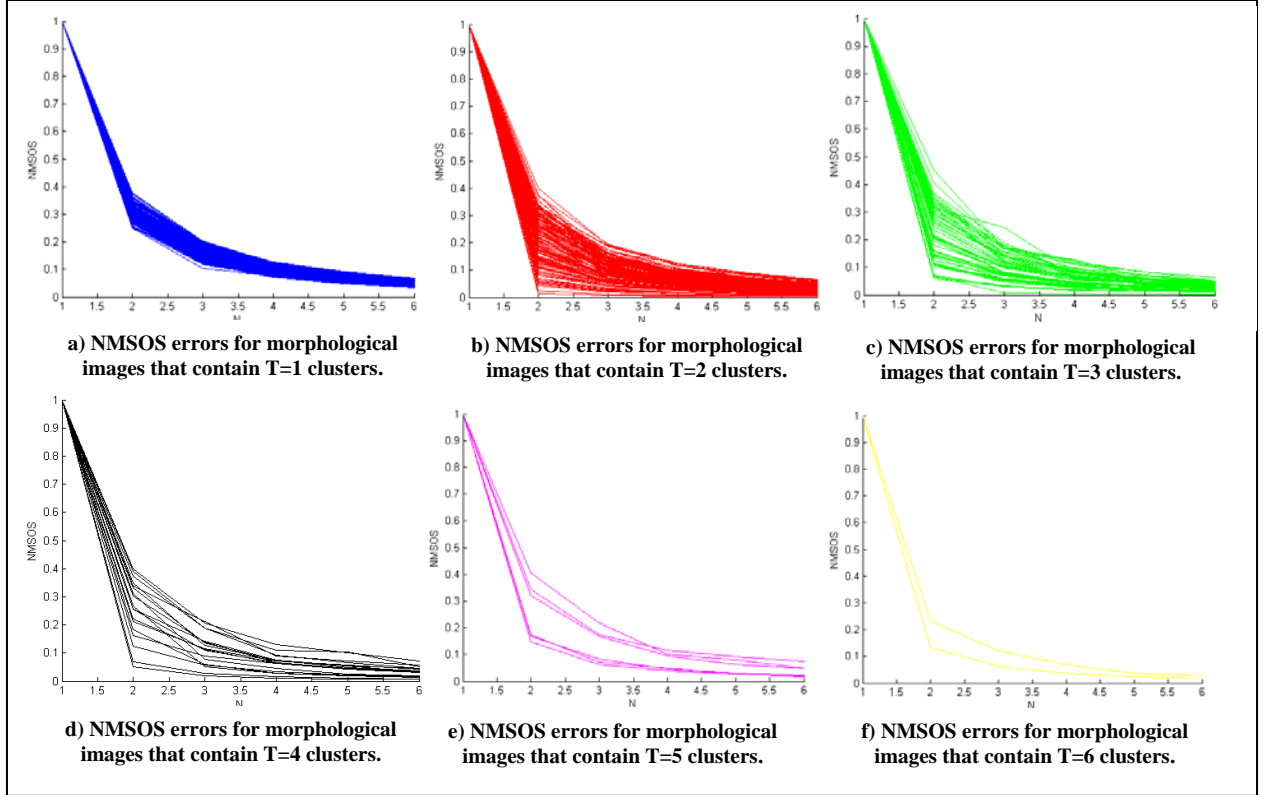f) NMSOS errors for morphological images that contain T=6 clusters.

Figure 13. 6 plots of the NMSOS errors organized for different values of *T*.

Table 2. Mean and standard deviation statistics of the NMSOS errors illustrated in figure 16.

T	$\bar{J}_1$ mean/std	$\bar{J}_2$ mean/std	$\bar{J}_3$ mean/std	$\bar{J}_4$ mean/std	$\bar{J}_5$ mean/std	$\bar{J}_6$ mean/std
1	1 / 0	0.299 / 0.023	0.156 / 0.019	0.100 / 0.014	0.072 / 0.010	0.055 / 0.008
2	1 / 0	0.222 / 0.089	0.102 / 0.044	0.061 / 0.027	0.042 / 0.019	0.032 / 0.014
3	1 / 0	0.230 / 0.101	0.108 / 0.050	0.062 / 0.029	0.041 / 0.019	0.030 / 0.014
4	1 / 0	0.256 / 0.106	0.115 / 0.059	0.063 / 0.032	0.047 / 0.027	0.033 / 0.018
5	1 / 0	0.260 / 0.109	0.129 / 0.065	0.075 / 0.034	0.054 / 0.029	0.040 / 0/022
6	1 / 0	0.183 / 0.071	0.092 / 0.041	0.054 / 0.027	0.031 / 0.009	0.022 / 0.006

Overlap exists between the NMSOS errors for different T. For example, consider the NMSOS error overlap between $T=1$ and $T=2$, as shown in figure 14. As is illustrated, the blue lines corresponding to $T=1$ are overlapped onto the red lines corresponding to $T=2$ and the choice for N_p is therefore unknown. This overlap condition signifies an ambiguity for selecting N_p when using the KP algorithm. This overlap is better recognized by further examination of the statistics in table 2. We will examine two stds (i.e., the 95th percentile) of the NMSOS error. For example, when $N=2$ for $T=1$, the mean is 0.299, the std is 0.023, two stds are 0.046, and the NMSOS error range is 0.253 to 0.345. The remaining NMSOS error ranges are shown in table 3. The NMSOS error ranges are also illustrated in figure 15. These results indicate that significant NMSOS error overlap exists for several values of N and the selection of N_p is therefore ambiguous.
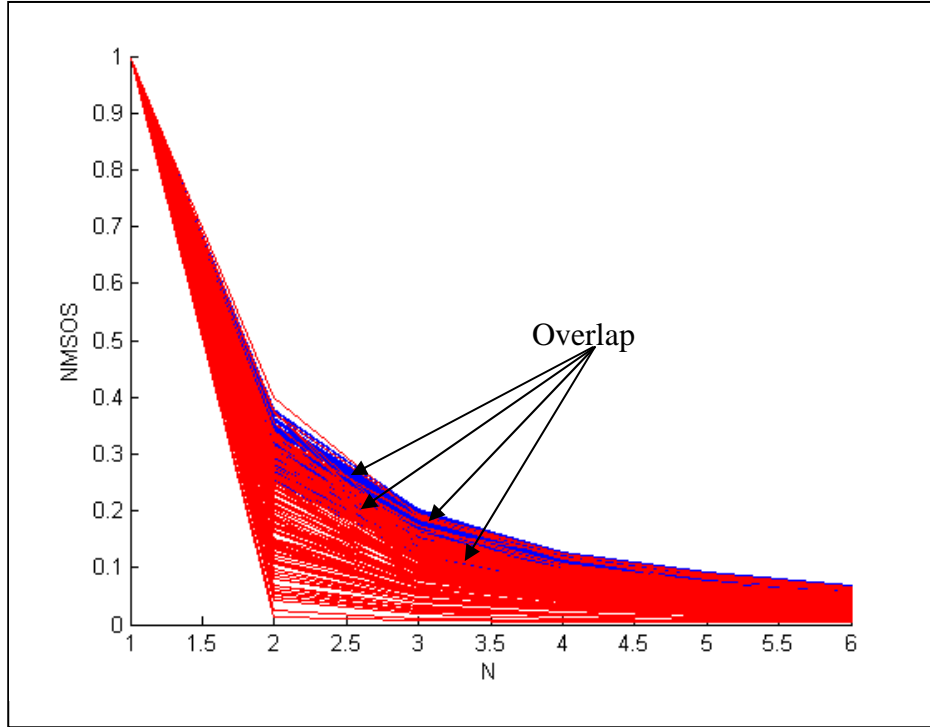

Figure 14. NMSOS errors for the $T=1$ (blue lines) and $T=2$ (red lines) frames. As is illustrated, the blue lines corresponding to $T=1$ are overlapped onto the red lines corresponding to $T=2$.

Table 3. NMSOS error range statistics.

|	*N=2*	*N=3*	*N=4*	*N=5*	*N=6*
T=1	0.253 to 0.345	0.117 to 0.194	0.072 to 0.128	0.051 to 0.092	0.039 to 0.070
T=2	0.044 to 0.401	0.014 to 0.189	0.007 to 0.115	0.005 to 0.080	0.004 to 0.060
T=3	0.027 to 0.433	0.007 to 0.209	0.004 to 0.120	0.004 to 0.079	0.003 to 0.057
T=4	0.044 to 0.468	–0.003 to 0.233	–0.001 to 0.127	–0.008 to 0.101	–0.003 to 0.068
T=5	0.042 to 0.479	–0.001 to 0.260	0.007 to 0.142	–0.004 to 0.111	–0.005 to 0.084
T=6	0.042 to 0.324	0.010 to 0.174	0.009 to 0.099	0.012 to 0.049	0.010 to 0.035
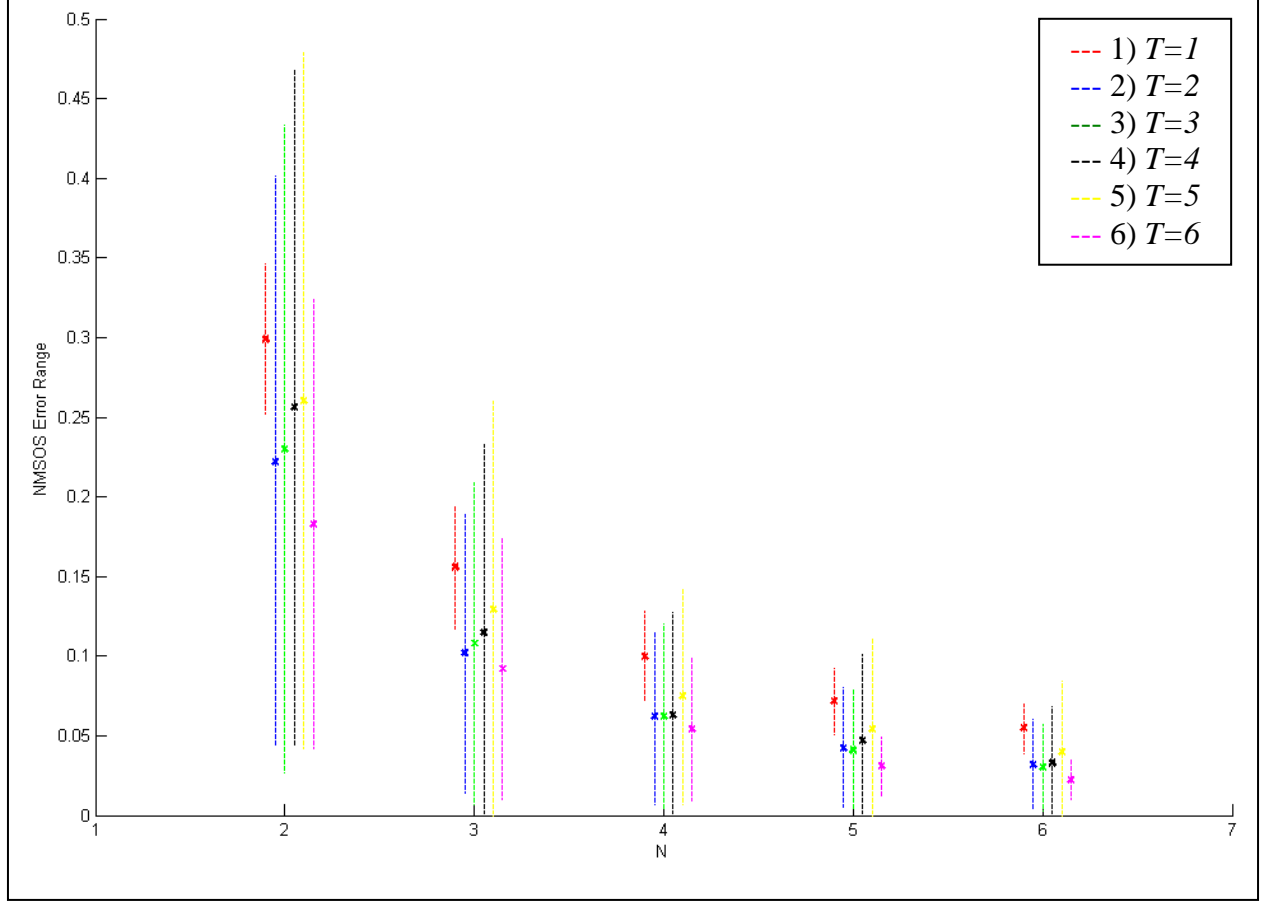
Figure 15. Graphical interpretation of the NMSOS error ranges.

Examples of morphological images that have overlapping NMSOS errors are shown in figure 16. The clusters/POIs contained in the morphological images are indicated by the red pixels. The white X's are used to indicate the centroid of each cluster. Consider figures 16(b)–(f) where the largest drop of the NMSOS error is at $N=2$, but more than two clusters exist in the morphological images. Examination of the NMSOS errors of figure 16(b) demonstrates a large drop at $N=2$ even though three clusters are present in the morphological image. Examination of the NMSOS errors of figure 16(c) demonstrates a very large drop at $N=2$ even though four clusters are present in the morphological image. Although the choice of N_P for these two examples is ambiguous, $N_P=2$ appears to be an appropriate selection given the proximity of the clusters in figures 16(b) and (c). For example, the top and bottom two clusters in figure 16(c) are very close, indicating that they could be merged.
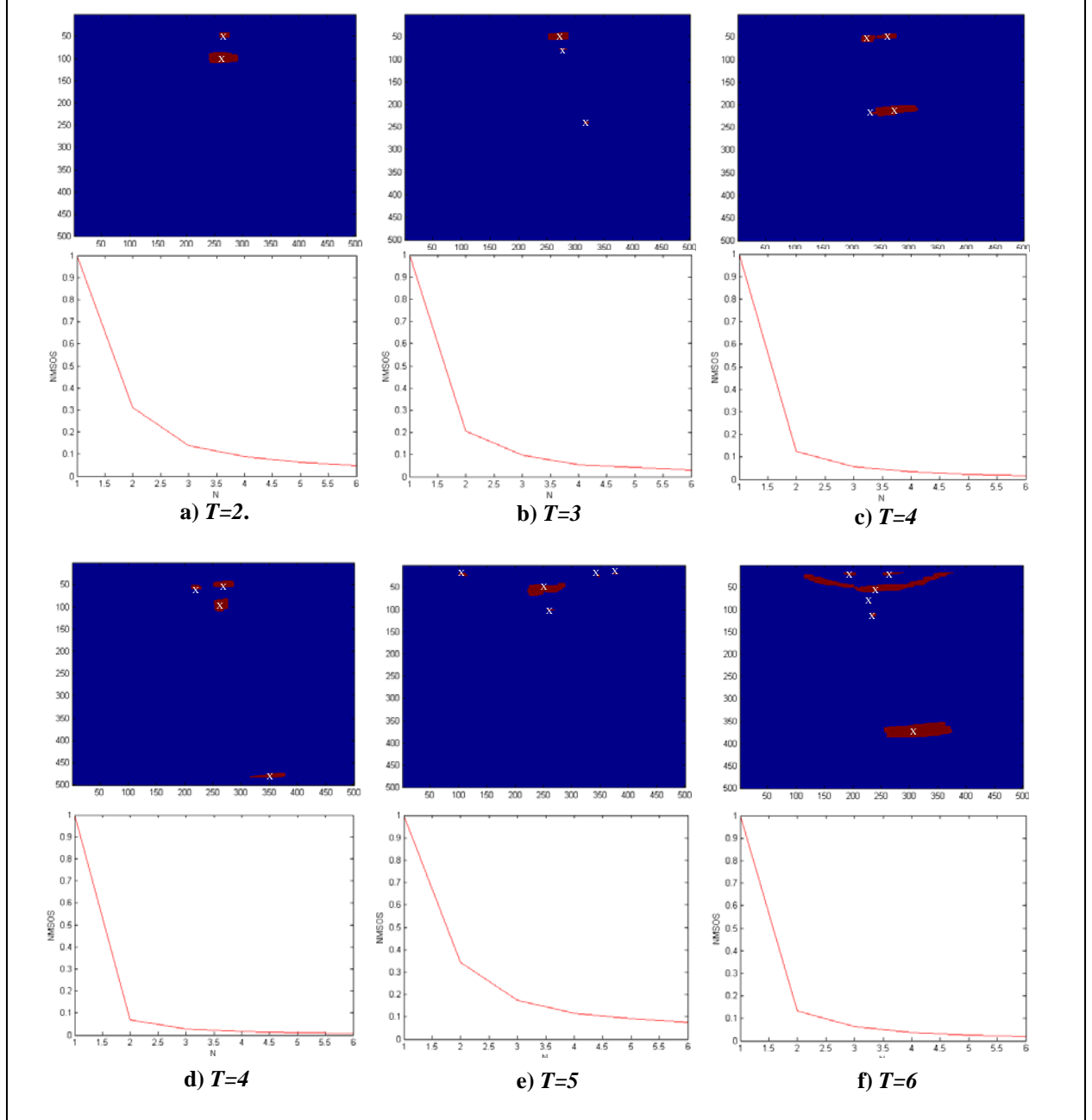
Figure 16. Examples of morphological images that have overlapping NMSOS errors. The clusters/POIs contained in the morphological images are indicated by the red pixels. The white X's are used to indicate the centroid of each cluster.

Examination of the NMSOS errors of figure 16(d) demonstrates a very large drop at $N=2$ even though four clusters are present in the morphological image. Examination of the NMSOS errors of figure 16(f) demonstrates another very large drop at $N=2$ even though six clusters are present in the morphological image. The clusters present in the morphological images of figures 16(d) and (f) appear large and separated, unlike the clusters in figure 16(c), and we would expect a significant drop at $N>2$. Here again, the selection of N_p is ambiguous.

17

Examination of the NMSOS errors of figure 16(a) and 16(e) indicates a drop at $N=2$ and $N=3$, similar to the NMSOS errors associated with $T=1$ (figure 13(a)); however, the corresponding morphological images contain more than one cluster. All images shown in figure 16 confirm that the choice of N_p is ambiguous and is due to the overlapping conditions of the NMSOS errors generated by the KP algorithm. Examination of other NMSOS errors and morphological images (other than the ones shown in figure 16) further substantiate that the choice of N_p is ambiguous. Based on these results, we would conclude that using NMSOS errors, generated by the KP algorithm, to identify N_p is insufficient for accurate identification of the true number of clusters present in the morphological images.

5.2 RPF Algorithm

The 12 data sequences of table 1 were also input into the RFP algorithm. The RFP algorithm processed the 12 data sequences and correctly identified the number of clusters present in the morphological images. In addition to identifying the number of clusters present in the morphological images, the RFP algorithm generated additional statistics for each T that are shown in table 4. Table 4 shows the mean number of pixels per cluster, mean cross-range cluster length measured in pixels, mean range cluster length measured in pixels, and the mean time to process a morphological image. The mean range cluster length is approximately the same for all T and is small relative to the mean cross-range cluster length. The mean range cluster length is small due to the high range resolution of the SIRE radar. The mean number of pixels per cluster is very large for $T=1$ and $T=6$ and is caused by the low cross-range resolution of the radar and the morphological processing algorithm. The morphological processing algorithm will merge false alarms together with the MT signature, resulting in a larger than expected clusters. An example of this effect is shown in figure 17, where the large clusters are a result of the several merged false alarms and MT signature.

Table 4. Cluster statistics generated by the RPF algorithm for morphological images.

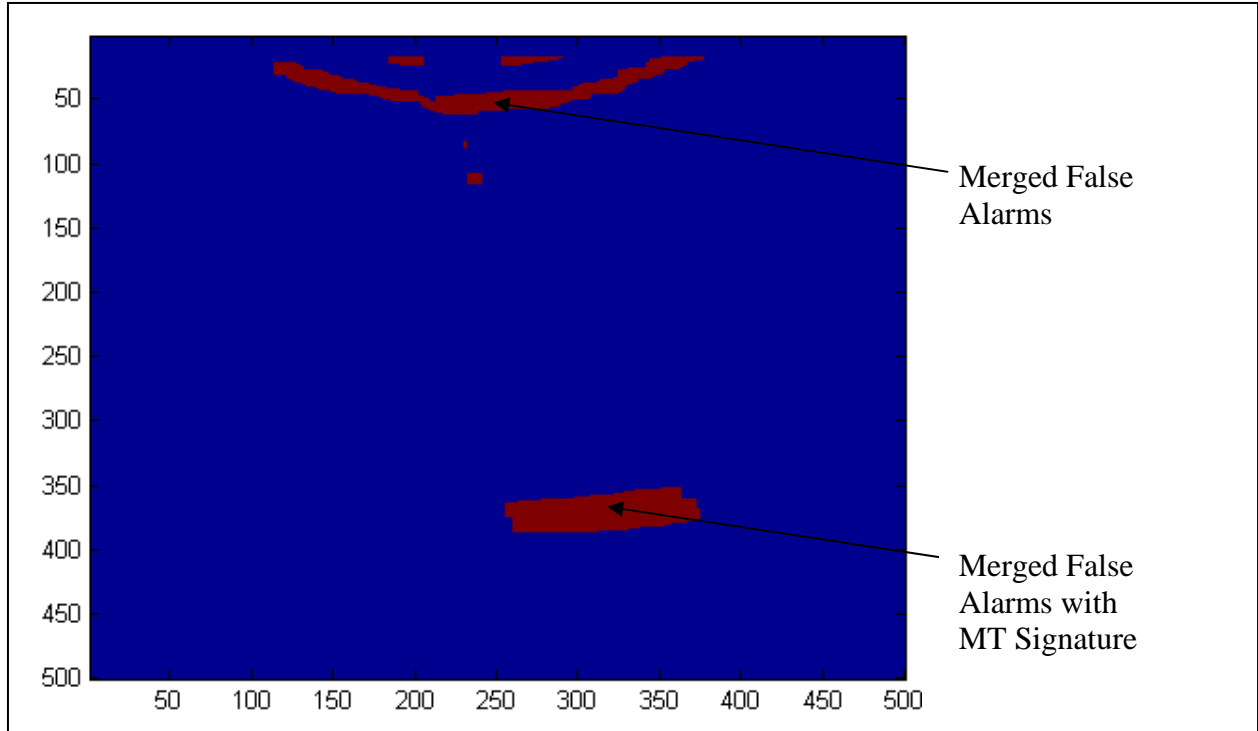|	Mean Number of Pixels Per Cluster	Mean Cross-Range Cluster Length (pixels)	Mean Range Cluster Length (pixels)	Mean Time to Process Image (s)
T=1	731.72	50.83	17.20	0.18
T=2	650.73	46.77	16.13	0.32
T=3	604.50	43.33	15.08	0.45
T=4	586.54	38.00	14.69	0.59
T=5	529.67	36.60	13.42	0.68
T=6	854.75	59.75	16.55	1.49
Overall	670.86	47.06	16.21	0.33

18

Figure 17. An example of larger than expected clusters when *T=6*. The large clusters are a result of the several merged false alarms and MT signature.

The computational complexity of the RPF algorithm was measured by examining the processing time of each morphological image. The mean time to process a morphological image is shown in table 4 and is less than 1 s except when *T=6*. Additional processing times are illustrated in figure 18. As is shown in this figure, 11 morphological images had a processing time greater than 1 s. Close examination of these morphological images shows that they contain clusters with a cluster size larger than 2000 pixels, which corresponds to over 2000 recursive function calls. The morphological image that took the longest time to process is shown in figure 17, where six clusters are present, and two out of the six clusters contain over 3000 pixels. These results indicate that the processing time of the RPF algorithm is dependent on the size of the clusters to be processed.
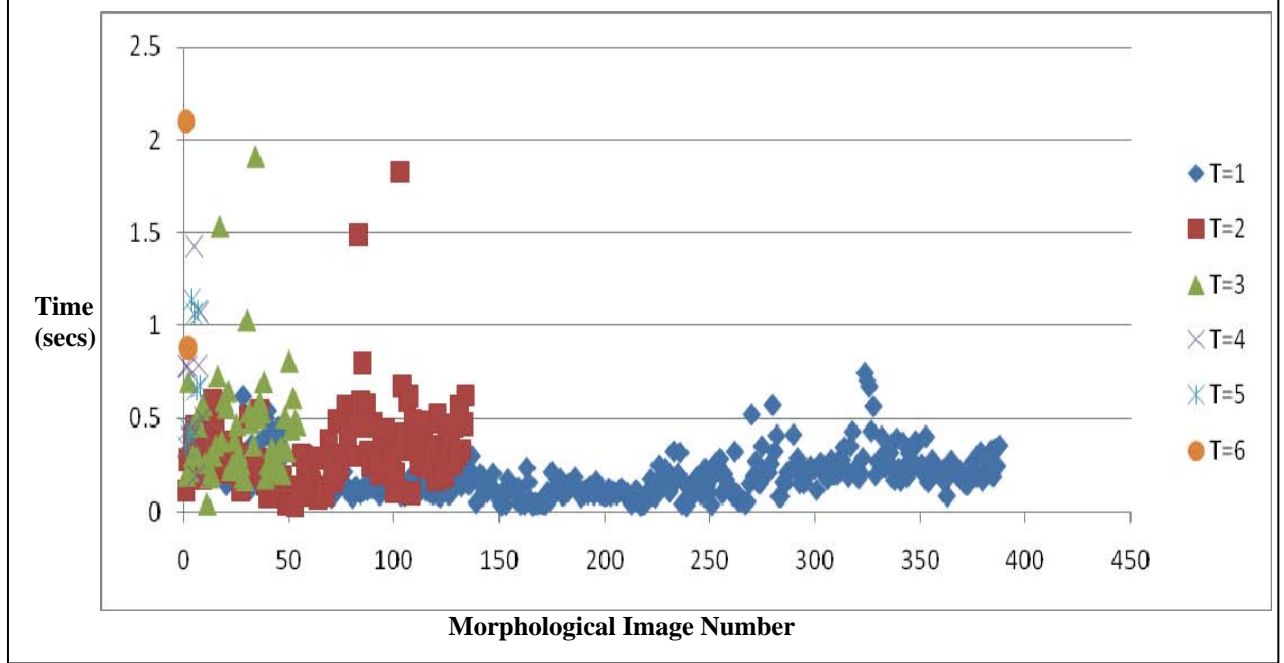
Figure 18. Processing times of the morphological images using the RPF algorithm.

As was demonstrated, the RPF algorithm is capable of correctly identifying the total number of clusters in the morphological images. It was shown that the computational complexity of the RPF algorithm is small and increases as the cluster size increases. One possible disadvantage of the RPF algorithm is that it does not have the capability to merge clusters in close proximity. For example, examination of figure 16(c) indicates that $T=4$, but $N_p=2$ is a good choice due to the proximity of the clusters. This merge feature could be easily added to the RPF procedure, but would require a redefinition of what we consider to be a cluster.

6.   Conclusions

In this report, we investigated the KP algorithm and the RPF algorithm to automatically identify the number of clusters to input into the k-Means algorithm. Both algorithms automate our MTI processing formulation and enable this formulation to be implemented in a real-time or near-real-time system. An advantage of the KP algorithm is its capability of merging clusters that are close. This was demonstrated by the example in figure 16(c), where $T=4$, but $N_p=2$ is a good choice due to the proximity of the clusters. A disadvantage of the KP algorithm is that it generates overlapping NMSOS for different T. Based on this result, we conclude that using NMSOS errors, generated by the KP algorithm, to identify N_p is insufficient for accurate identification of the true number of clusters present in the morphological images. The advantages of the RPF algorithm is its capability of correctly identifying the "true" number of clusters in all morphological images and its low computational complexity. One possible disadvantage of the

20

RPF algorithm is that it does not have the capability to merge clusters in close proximity. This merge feature could be easily adapted to the RPF procedure, but would require a redefinition of what we consider to be a cluster. We, therefore, conclude that the RPF algorithm outperforms the KP algorithm for accurate identification of the true number of clusters present in the morphological images.

7. References

1. Farwell, M.; Ross, J.; Lutrell, R.; Cohen, D.; Chin, W.; Dogaru, T. Sense Through the Wall System Development and Design Considerations. *J. of the Franklin Institute* **Sept 2008,** *345* (6), 570–591.

2. Martone, A.; Ranney, K.; Innocenti, R. Through the Wall Detection of Slow Moving Personnel. *Proceedings of the SPIE Conference on Radar Sensor Technology XIII*, vol. 7308, Orlando, Fl, April 2009.

3. Martone, A.; Ranney, K.; Innocenti, R. *Moving Target Indication for Transparent Urban Structures*; ARL-TN-4809; U.S. Army Research Laboratory: Adelphi, MD, May 2009.

4. Novak, L. Change Detection for Multi-polarization, Nulti-pass SAR. *Proceedings of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery XII*, vol. 5808, Orlando, FL, March 2005, 234–246.

5. Ressler, M.; Nguyen, L.; Koenig, F.; Wong, D.; Smith, D. G. The Army Research Laboratory (ARL) Synchronous Impulse Reconstruction (SIRE) Forward-Looking Radar. *Proceedings of the SPIE Conference on Unmanned Systems Technology IX,* vol. 6561, Bellingham, WA, April 2007, 656105-1–656105-12.

6. Nguyen, L.; Wong, D.; Ressler, M.; Koenig, F.; Stanton, B.; Smith, G.; Sichina, J.; Kappra, K. Obstacle Avoidance and Concealed Target Detection Using the Army Research Lab Ultra-Wideband Synchronous Impulse Reconstruction (UWB SIRE) Forward Imaging Radar. *Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets XII,* vol. 6553, Bellingham, WA, April 2007, 65530H-1–65530H-8.

7. McCorkle, J. Focusing of Synthetic Aperture Ultra Wideband Data. *Proceedings of the IEEE International Conference on Systems Engineering*, Dayton, OH, August 1991, 1–5.

8. Nguyen, L. *Image Resolution Computation for Ultra-Wideband (UWB) Synchronous Impulse Reconstruction (SIRE) Radar*; ARL-TN-0294; U.S. Army Research Laboratory: Adelphi, MD, July 2007.

9. Gandhi, P. P.; Kassam, S. A. Analysis of CFAR Processors in Homogeneous Background. *IEEE Transactions on Aerospace and Electronic Systems* **July 1988**, *24* (4), 427–445.

10. Jain, A. *Fundamentals of Digital Image Processing*; Prentice-Hall International, Englewood Cliffs, NJ, 1989.

11. Wilpon, J.; Rabiner, L. A Modified K-means Clustering Algorithm for Use in Isolated Work Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* **July 1985**, *33* (3), 587–594.

12. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the 5th Berkeley Symposium on Probability and Statistics*, vol. 1, Berkeley, CA, 1967, 281–296.

13. Blackman, S.; Popoli, R. *Design and Analysis of Modern Tracking Systems*; Artech House: Norwood, MA, 1999.

14. Duta, R.; Hart, P.; Stork, D. *Pattern Classification*; 2nd ed. John Wiley and Sons Inc.: New York, NY, 2001.

15. Thorndike, R. Who Belongs in the Family?" *Psychometrika* **Dec 1953**, *18* (4), 267–276.

16. Zhao, Q.; Xu, M.; Franti, P. Knee Point Detection on Bayesian Information Criteria. *Proceedings of the 20ᵗʰ IEEE International Conference on Tools with Artificial Intelligence*, vol. 2, November 2008, 431–438.

List of Symbols, Abbreviations, and Acronyms

ARL   U.S. Army Research Laboratory

ATD   automatic target detection

CD   change detection

CFAR   constant false alarm rate

KP   knee-point

MT   moving target

MTI   moving target indication

NMSOS   normalized minimized SOS

POIs   points of interest

ROI   region of interest

RPF   recursive pixel finding

SAR   synthetic-aperture-radar

SIRE   Synchronous Impulse Reconstructive

SNR   signal-to-noise ratio

SOS   sum of squares

STTW   sensing through the wall

UWB   ultrawideband

INTENTIONALLY LEFT BLANK.